REMARKS

The Office Action fails to show that claims 1-16 are anticipated under 35 USC §102(b) by US patent number 5,276,881 to Chan et al. (hereinafter "Chan"). The rejection is respectfully traversed because the Office Action fails to show that all the limitations of the claims are identically taught by Chan.

As to claims 1 and 16, the limitations relate to switching between multiple implementations of a routine in a library of routines that are linked with an application program. The limitation include compiling a plurality of implementations of a routine into respective object code modules, the routine having an associated name and each implementation adapted to a selected hardware configuration; associating the object code modules with the name of the routine and respective sets of hardware characteristics; and resolving when the application program is loaded into memory of the computer system, a reference to the routine using the sets of hardware characteristics and a hardware configuration of the system. The rejection fails to show that Chan identically teaches these limitations.

For example, the Office Action fails to show that Chan teaches compiling a plurality of implementations of a routine into respective object code modules. The cited section portion of Chan (col. 60, ll. 30-35) briefly mentions generating object code from input machine instructions and storing the object code in bit patterns expected by the hardware. Furthermore, from Chan's FIG. 2, it appears that a single routine implementation may be compiled into multiple modules of object code, each module suitable for a particular hardware platform (FIG. 2). The Office does not show, nor does it appear that Chan teaches compiling a plurality of implementations of a single routine into respective code modules. Further clarification is respectfully requested if others of Chan's teachings are thought to show the claimed compilation of a plurality of implementations of a routine into respective object code modules.

The Office Action further fails to show object code modules being associated with the routine's name and respective sets of hardware characteristics. The cited portion of Chan teaches:

> The object File Generator 1358 writes other required information to the Object file 1362 according to the format expected by the native linker and

> loader. This might include Symbol Table 1312 and Type Table 13006 information, relocation information, and object file management information. (col. 60, ll. 35-40).

This portion of Chan in no apparent way teaches the claimed association of the object code modules with respective sets of hardware characteristics. Furthermore, Chan appears to not need the claimed association since Chan generates object code on specific platforms from a compiler intermediate representation (FIG. 2, col. 12, ll. 17-39). Further clarification is respectfully requested if others of Chan's teachings are thought to show these limitations.

The Office Action also fails to show that Chan identically teaches resolving when the application program is loaded into memory of the computer system, a reference to the routine using the sets of hardware characteristics and a hardware configuration of the system. Chan's col. 55, ll. 60-65 is cited as teaching these limitations. However, the cited portion of Chan teaches:

> The HPcode-Plus Object files 1160 deposited into the HPcode-Plus archive file are extracted, installed, and archived into a standard library file at the target computer platform 216, 226. The resulting library file is then searched as usual for resolving references by a native linker on the install site 216, 226. All system specific libraries on the native computer platform 206 used by computer programs 202 must be translated into HPcode-Plus archive files and provided to the target computer platforms 216, 226. (col. 55, ll. 60-65).

It appears that this section of Chan refers to installing and linking a library on specific computer platforms. It will be appreciated that the claims relate to resolving references when the application program is loaded into memory, which is thought to be different form installing. In addition, there appears to be no mention of resolving the references using the associated sets of hardware characteristics. The cited text simply mentions "searching as usual for resolving references by a native linker..." Further clarification is respectfully requested if others of Chan's teachings are thought to identically show these limitations.

Claim 2 includes further limitations related to establishing a symbol table having a plurality of entries, each entry including a name of a routine and a reference to an object code module in the library. The Office Action cites Chan's FIG. 13, elements 1308 and 1310, along with Chan's col. 60, ll. 35-40, as identically teaching these limitations. However, it is not apparent which of Chan's teachings are thought to identically teach the symbol table having a reference to an object code module in the library as claimed, even though a "symbol

table" is referenced. Further clarification is respectfully requested if others of Chan's teachings are thought to identically show these limitations.

Claim 3 includes limitations that relate to adding a plurality of entries to the symbol table and associating respective sets of hardware characteristics with the plurality of entries for the routine having a plurality of implementations. The Office Action cites Chan's col. 48, ll. 1-15 as teaching these limitations. However, this text mentions creating a symbol table having symbolic identifiers, symbolic information, and symbolic kind information. There is no apparent teaching or even a suggestion of entries in the symbol table being associated with sets of hardware characteristics. Further clarification is requested if the rejection is maintained.

Claim 4 which depends from claim 3 and claim 7 which depends from claim 1, include limitations that relate to the hardware characteristics including at least one of clock speed of the processor, processor model, cache configuration of the system, hardware operation latency times, instruction set characteristics, bypass characteristics, branch prediction behavior, pre-fetching capability, information describing stall conditions, branch penalties, size and associativity of processor data structures, queue sizes for out-of-order or decoupled processors, and the number of processors in a multi-processor system. The Office Action cites Chan's col. 54, ll. 60-65 as teaching these limitations. However, this text teaches a high-level optimizer that computes performance improving transformations on the instructions. No mention is made of any association of any hardware characteristics with object code modules or with entries in the symbol table. Further clarification is requested if the rejection is maintained.

Claims 5, 6, and 8 include limitations that relate to the resolving step further comprising obtaining the hardware configuration of the system from at least one of a system configuration data file, one or more system identification registers, and system firmware. The Office Action cites Chan's col. 58, ll. 18-23 as teaching these limitations. However, this text simply references a machine configuration file containing architecture specific register sizes, co-processor availability, and instruction cycle times. Chan's preceding paragraphs explains that this information is used by a low-level code generator to select a sequence of instructions to implement a quadruple from the intermediate code. This clearly does not identically teach resolving a reference to a routine by using a configuration data file, system identification registers, or system firmware when the program is loaded.

Claims 9-15 include limitations similar to those of claims 1-8 and are allowable for at least the reasons set forth above. In addition, the Office Action fails to show that Chan identically teaches the further refinements set forth in claims 1-9 of the limitations of claims 1-8 and the additional processing set forth in claims 1-9 related to the limitations of claims 1-8. Since the Office Actions fails to show that claims 1-8 are anticipated by Chan, the Office Action also fails to show that Chan teaches the further refinements and further processing as set forth in claims 9-15.

As explained above, the Office Action fails to show that Chan identically teaches all the limitations of claims 1-16. Therefore, the rejection is improper and should be withdrawn. If the rejection is maintained, further explanation is requested to address the deficiencies in the Office Action.

Withdrawal of the rejection and reconsideration of the claims are respectfully requested in view of the remarks set forth above.

No extension of time is believed to be necessary for consideration of this response. However, if an extension of time is required, please consider this a petition for a sufficient number of months for consideration of this response. If there are any additional fees in connection with this response, please charge Deposit Account No. 08-2025 (HPCO.008PA).

Respectfully submitted,

CRAWFORD MAUNU PLLC
1270 Northland Drive, Suite 390
Saint Paul, MN 55120
(651) 686-6633

By: _____
Name: LeRoy D. Maunu
Reg. No.: 35,274